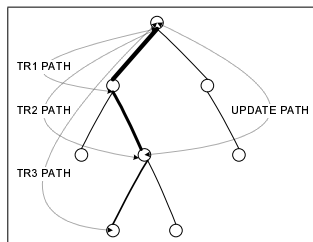


SYRCoDIS'2005 Proceedings:
Polygraphy Corrections

Volume A



Page 4: Correct picture looks like:

Page 42:

Correct code looks like:

```
foreach (Value v in RSIF)
{
  lRv, lSv : ref to list of RID;
  Initialize lRv and lSv;
  foreach (tR.rid in ↑lRv)
  {
    Lr, result : ref to list of RID;
    Lr = Workmapp.Get(tR.rid); (!)
    if (Lr = NULL)
      result = lSv;
    else
      result = Intersect(↓Lr, ↑lSv); (!)
    Workmapp.Put(tR.rid, result);
  }
}
```

Pages 58,59,60,61:

Underbar symbol «_» in the code is printed under the next symbol after underbar.

Volume B

Page 4:

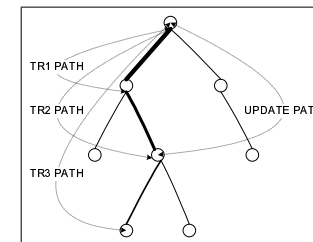
Correct references look like:

[2] Claudia Antunes and Arlindo Oliveira. Temporal Data Mining: an Overview. In *KDD Workshop on Temporal Data Mining*, pages 1–13, San Francisco, 2001.

[4] F. Hoppner. Learning Temporal Rules from State Sequences. In *WLTS*, pages 25–31, Seattle, USA, 2001.

SYRCoDIS'2005 Proceedings:
Polygraphy Corrections

Volume A



Page 4: Correct picture looks like:

Page 42:

Correct code looks like:

```
foreach (Value v in RSIF)
{
  lRv, lSv : ref to list of RID;
  Initialize lRv and lSv;
  foreach (tR.rid in ↑lRv)
  {
    Lr, result : ref to list of RID;
    Lr = Workmapp.Get(tR.rid); (!)
    if (Lr = NULL)
      result = lSv;
    else
      result = Intersect(↓Lr, ↑lSv); (!)
    Workmapp.Put(tR.rid, result);
  }
}
```

Pages 58,59,60,61:

Underbar symbol «_» in the code is printed under the next symbol after underbar.

Volume B

Page 4:

Correct references look like:

[2] Claudia Antunes and Arlindo Oliveira. Temporal Data Mining: an Overview. In *KDD Workshop on Temporal Data Mining*, pages 1–13, San Francisco, 2001.

[4] F. Hoppner. Learning Temporal Rules from State Sequences. In *WLTS*, pages 25–31, Seattle, USA, 2001.

Page 43:

Correct code looks like:

```
foreach (PartBound p in PartBounds)
{
  Workmapp = map : RID -> ref to list of RID;
  foreach (Value v in RSIF)
  {
    lRv, lSv, lRpv : ref to list of RID;
    Initialize lRv and lSv;
    lRpv = Select(↑lRv, p);
    foreach (tR.rid in ↑lRpv)
    {
      Lr, result : ref to list of RID;
      Lr = Workmapp.Get(tR.rid);
      if (Lr = NULL)
        result = lSv;
      else
        result = Intersect(↓Lr, ↑lSv);
      Workmapp.Put(tR.rid, result);
    }
  }
}
```

Page 43:

Correct code looks like:

```
foreach (PartBound p in PartBounds)
{
  Workmapp = map : RID -> ref to list of RID;
  foreach (Value v in RSIF)
  {
    lRv, lSv, lRpv : ref to list of RID;
    Initialize lRv and lSv;
    lRpv = Select(↑lRv, p);
    foreach (tR.rid in ↑lRpv)
    {
      Lr, result : ref to list of RID;
      Lr = Workmapp.Get(tR.rid);
      if (Lr = NULL)
        result = lSv;
      else
        result = Intersect(↓Lr, ↑lSv);
      Workmapp.Put(tR.rid, result);
    }
  }
}
```